(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
27. Dezember 2001 (27.12.2001)

PCT

(10) Internationale Veröffentlichungsnummer
WO 01/98899 A2

(51) Internationale Patentklassifikation[7]: G06F 9/445

(21) Internationales Aktenzeichen: PCT/CH01/00373

(22) Internationales Anmeldedatum:
15. Juni 2001 (15.06.2001)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
1222/00 20. Juni 2000 (20.06.2000) CH

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: SYSFORMANCE AG [CH/CH]; Badenerstrasse 281, CH-8003 Zürich (CH).

(72) Erfinder; und

(75) Erfinder/Anmelder *(nur für US)*: FISCHER, David [CH/CH]; Mühlemattstrasse 61, CH-3007 Bern (CH).

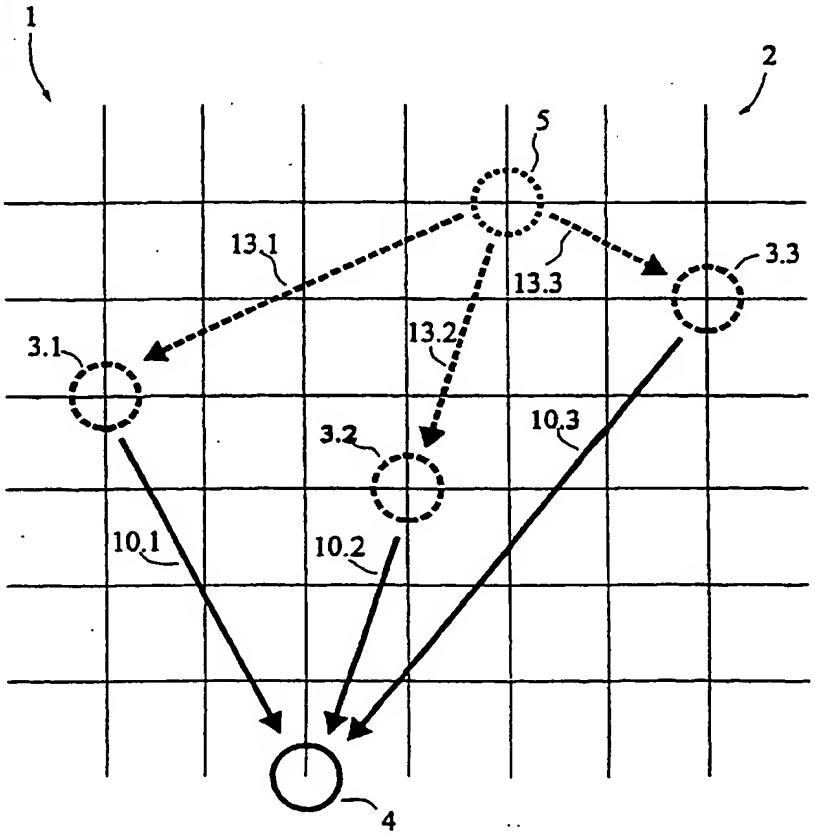
(74) Anwalt: FREI PATENTANWALTSBÜRO; Postfach 768, CH-8029 Zürich (CH).

(81) Bestimmungsstaaten *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Fortsetzung auf der nächsten Seite]

(54) Title: SERVER MONITORING

(54) Bezeichnung: SERVERÜBERWACHUNG



(57) Abstract: The invention relates to a method for running a plug-in on one or more computers, especially for monitoring purposes. The plug-in is transmitted to at least one computer (11.1, 11.2, 11.3) via a network (2). Afterwards, the plug-in prompts the at least one computer (11.1, 11.2, 11.3) to run this plug-in.

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren um ein Plugin auf einem oder mehreren Computer, insb. zu Überwachungszwecken, zur Ausführung zu bringen. Das Plugin wird über ein Netzwerk (2) an mindestens einen Computer (11.1, 11.2, 11.3) übermittelt. Anschliessend veranlasst das Plugin den mindestens einen Computer (11.1, 11.2, 11.3), dieses zur Ausführung zu bringen.

WO 01/98899 A2

(84) **Bestimmungsstaaten** *(regional)*: ARIPO-Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI-Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

SERVERÜBERWACHUNG

Die vorliegende Erfindung liegt auf dem Gebiet der Netzwerk-, resp. Internettechnologie. Die Aufgabe wird durch die in den Patentansprüchen definierte Erfindung gelöst.

Heute hat sich besonders das Internet als weltweites Kommunikationsmittel etabliert. Die Qualität der angebotenen Dienste spielt daher eine wesentliche Rolle. Firmen die auf dem Internet ihre Dienstleistungen anbieten, haben ein grosses Interesse, dass ihre Server einwandfrei funktionieren und dass unberechtigte Zugriffe frühzeitig erkannt und Massnahmen ergriffen werden können. Eine Überwachung dieser Dienste ist bis heute nicht bekannt. Aus diesem Grund werden viele auf dem Internet angebotene Dienstleistung nicht oder ungenügend in Anspruch genommen. Die Dienste weisen häufig ungenügende Qualität (zu lange Antwortzeiten, usw.) auf, was die potentiellen Benutzer davon abhält. Unberechtigte Zugriffe und Veränderungen werden in der Regel nur sehr schlecht und mit Verzögerung erkannt. Dies führt dazu, dass schädliche Software wie Viren, usw. sich unbemerkt über längere Zeiträume ausbreiten können. Schäden weltweit in Milliardenhöhe sind keine Seltenheit.

Es ist Aufgabe der vorliegenden Erfindung ein Verfahren zur Ausführung von Plugins zu zeigen; insbesondere zur Überwachung von Netzwerken, Internetdienstleistungen und Servern.

Die Idee der hier offenbarten Erfindung basiert u.a. darauf, eine Proxy-Server Unterstützung des Internet HTTP-Protokolls zum Zweck der automatischen Aufzeichnung und der späteren automatischen Wiederabspielung eines Datenverkehrs von einem oder mehreren HTTP-Clients (z.B. Web-Browsern), die mit einem HTTP-Server oder HTTP-Proxy-Server kommunizieren, zu verwenden. Vorzugsweise referentiell aufgezeichnete Daten werden dabei in einer Form gespeichert, die es erlaubt den vollständigen Datenverkehr der vom Client und vom Server generiert wird (Requests), zu einem späteren, bestimmbaren Zeitpunkt automatisch und beliebig oft, insb. von verschiedenen geographischen Orten aus und unter Einhaltung von definierten Kriterien zu wiederholen, zu überwachen und auszuwerten. Der Vorgang erfolgt in der Regel ohne Zutun des ursprünglichen, generierenden Clients. Bei der Aufzeichnung eines Datenverkehrs werden üblicher Weise auch die Antwort-Daten des Servers (Responses) ganz oder teilweise aufgezeichnet. Dadurch ist es erstmals möglich, dass bei einer späteren Anwendung der aufgezeichneten Client-Requests kontrolliert werden kann, ob der Server analoge, gleichbleibende Daten liefert, oder ob er von einer definierten Norm abweicht. Dies spielt zur periodischen Überwachung von unberechtigten Zugriffen eine relevante Rolle.

Im Zusammenhang mit der Überwachung z.B. von Viren wird bei Bedarf anstelle einer meist erfolglosen Suche nach schädlichen Programmen, die Information periodisch mit gesicherten und vertrauenswürdigen Referenzdaten (von einem oder mehreren entfernten Standpunkten aus) verglichen. Eine entsprechender Vergleich liefert aussagekräftige Daten mit minimalem Aufwand. So ist es z.B. möglich, dass eine Firma entsprechende Dienste anbietet indem Sie Referenzdaten von einzelnen Servern periodisch mit deren momentanen Verhalten, z.B. zwecks Qualitätssicherung, vergleicht und überwacht. Bei Bedarf werden die Antwortzeiten des Servers aufgezeichnet. Die Überwachung erfolgt vorteilhafter Weise von verschiedenen geografischen Orten aus, derart, dass eine Überwachung über mehrere Kanäle erfolgt. Damit ist es zudem möglich die Performance und Abweichungen derselben von definierba-

ren Grenzwerten (insb. über unterschiedliche Wege) zu vergleichen und auszuwerten. Entsprechend Alarmmeldungen werden falls erforderlich abgesetzt.

Der Inhalt des Datenverkehrs über ein gewähltes Protokoll (bspw. HTTP) spielt beim hier beschriebenen Verfahren eine eher untergeordnete Rolle, d.h. es können sämtliche Inhalte aufgezeichnet und wieder abgespielt werden, auch wenn diese z.B. Inhalte von höher liegenden Protokollen, wie z.B. JavaScript oder SSL, betreffen. Weitere Anwendungsbeispiele des hier beschriebenen Verfahrens sind, z.B. das Aufzeichnen von interaktiven Webbrowser Surfsessions. Dabei ist es vorteilhaft eine Referenz-Session aus einer oder mehreren Sessions zu generieren. Eine Auswertung und ein späteres Anwenden dieser Surf-Sessions z.B. in Form von Last-Test-Routinen dient der referentiellen Überwachung und der Kontrolle von unberechtigten Zugriffen, sowie der Performancemessung. Insbesondere wird auch die Verfügbarkeit des Servers überwacht, um Hardwaredefekte oder Abstürze zu überwachen. Ein Vergleich der referenzierten (aufgezeichneten) Server-Antwortdaten mit dem bei einer Anwendung derselben auf einen Server, insb. über mehrere Kanäle oder Pfade, generierten Datenverkehr wird bevorzugt als Mechanismus zur Erkennung von Modifikationen des Dateninhalts des Servers sowie zur ortsabhängigen Performancemessung eingesetzt. Illegale Zugriffe und Veränderungen werden damit zuverlässig und schnell erkannt.

Normalerweise sind die z.B. im heute weit verbreiteten HTTP-Protokoll vorgesehenen Einsatzzwecke von Proxy-Servern u.a. die folgenden: Zwischenspeichern von Daten, zum Zweck der Verkürzung der Antwortzeiten; Protokollierung und Auswertung des Datenverkehrs zwischen Client und Server, im Hinblick auf die Kontrolle des Surf-Verhaltens individueller natürlicher Personen (Beobachtung und Kontrolle der Person, Unterdrückung unerwünschter Websites etc.); Unterbindung der direkten Verbindung einzelner Computer von Endbenutzern mit dem Internet aus Sicherheitsgründen. Die hier offenbarte Erfindung basiert in entfernter Weise auf der

Funktionalität eines Proxy-Servers auf. Im Unterschied hierzu wird die eigentliche Hauptfunktion eines herkömmlichen Proxy-Servers dabei nicht oder nur in nebensächlicher Weise verwendet. Die hier offenbarte Erfindung weist zu einem herkömmlichen Proxy-Server u.a. die folgenden Unterschiede auf:

- Damit alle Dateninhalte zwischen Client und Server werden (zeitlich) kompakt aufgezeichnet werden können, werden bei der Erfindung sämtliche Cache- Mechanismen (sowohl des normalen HTTP-Protokolls sowie auch des HTTP-Proxy-Protokolls, insb. der direkt dargestellten und der vom Client ausgeführten Referenzen) ausser Acht gelassen (bei Bedarf kann eine Verwendung vorgesehen werden) oder unterdrückt. Die Erfindung erfordert daher in der Regel keinen eigenen Cache.

- Insbesondere werden gezielt alle Informationen des Clients an den Server und des Servers an den Client über Cache-Möglichkeiten unterdrückt, um zu erreichen, dass alle relevanten Daten übermittelt werden.

Die Erfindung weist Mittel zur Aufzeichnung auf. Mit eigens dafür vorgesehenen Schnittstellen werden diese Mittel gesteuert ("Start Record"). In diesem Zustand werden alle Requests/Responses in einer definierten Datenstruktur gespeichert so, dass der Verlauf derselben zu einem späteren Zeitpunkt mit entsprechenden Mittel (beispielsweise einer entsprechend programmierten Maschine) nachvollzogen werden können. Die z.B. referentiell aufgezeichneten Daten werden mit Vorteil in einer entsprechenden Bibliothek angelegt.

Aus den aufgezeichneten Daten werden bei Bedarf automatisch oder manuell erfindungsgemässe Plugins erzeugt (vgl. hierzu weiter unten), die über erfindungsgemä-

sse Mittel, z.B. Sonden (vgl. hierzu weiter unten), ausführbar sind, derart dass insb. von unterschiedlichen Orten aus der selbe Test gleichzeitig durchführbar ist. Damit ist es möglich einen Server mit verschiedenen oder mehrere Server mit speziellen Referenzdaten zu überwachen. Die Erfindung lässt sich auf nur einen Client gezielt oder aber auf alle Clients anwenden. Bei Clients mit einer separaten Aufzeichnung wird vorteilhafter Weise vom Client eine HTTP-Authentification verlangt. Diese kann bei jedem Request eines Clients danach dazu benutzt werden, um die Aufzeichnungsdaten der einzelnen Clients einzeln zu führen.

Es versteht sich von selber, dass die Erfindung falls erforderlich auch HTTP zu HTTPS (SSL) Konvertierungen bzw. höher liegende Protokolle unterstützen kann. Beispielsweise können, zum Aufzeichnen von HTTPS-Abfragen, vom Client unverschlüsselte Anfragen an den Server gemacht werden. Diese unverschlüsselten Anfragen werden dann erst durch die Erfindung verschlüsselt und an den Server weitergeleitet. Die Antwort wird wiederum durch die Erfindung entschlüsselt und an den Client zurück geleitet. Dabei ist es besonders vorteilhaft, dass das SSL-Protokoll durch die Erfindung entschlüsselt wird und nicht erst durch den Client. Dadurch ist es möglich, den Datenaustausch zwischen Client und Server auch bei einer Verschlüsselung aufzuzeichnen. Höherliegende Protokolle werden zum Zweck der Aufzeichnung/Überwachung gezielt aufgebrochen, indem Anstelle eines vorgesehen Tunnelling-Verfahrens eine Client-Server-Client-Server Verfahren vorgesehen wird.

Aus dem Stand der Technik sind Plugins bekannt. Plugins sind typischerweise universell einsetzbare Programme, die darauf spezialisiert sind, irgendeine Funktion auszuführen. Um ein Plugin zu aktivieren ist eine entsprechende Plugin-Schnittstelle erforderlich. Bei Java-Programmen beispielsweise erfolgt dies über ein entsprechendes Interface. In der Regel ist es so, dass ein Plugin aufgrund einer Anfrage bzw. eines Bedarfs eines Programms geladen wird (z.B. von einem Web-Browser). Sowohl bei CORBA als auch bei RMI (Java Remote Method Invocation) werden aber,

im Unterschied zur hier offenbarten Erfindung, nur Daten, bzw. Variablen ausgetauscht, es wird jedoch kein Programmcode übermittelt. Bei den erfindungsgemässen Plugins wird im Unterschied zum Stand der Technik typischerweise der Programm-Code übertragen. Bei konventionellen Plugins geht zudem der Anreiz zum Laden eines Plugins immer von dem Ort aus, an dem das Plugin auch ausgeführt wird (von innen). Bei erfindungsgemässen Plugins kommt dieser Anreiz jedoch von einem anderen Ort, also typischerweise von aussen.

Die erfindungsgemässen Plugins funktionieren vorteilhafter Weise wie folgt: An einem ersten Ort (Ausgangsort) wird zu einem bestimmten Zeitpunkt veranlasst, dass ein Plugin an einem zweiten Ort (Zielort) mittels einem geeigneten Mittel ausgeführt werden soll. Das Plugin wird darauf an den zweiten Ort (Zielort) mit einer Aufforderung zur Ausführung übertragen. Das Resultat besteht also darin, dass am zweiten Ort (Zielort) ein Plugin ausgeführt wird welches z.B. ein Resultat an den ersten Ort (Ausgangsort) zurückgemeldet. Einzige Anforderung am zweiten Ort (Zielort) ist, dass erfindungsgemässe Plugins empfangbar, resp. ausführbar (= „anspringen") sind. Es ist nicht erforderlich, dass der Zielort über den Inhalt des erfindungsgemässen Plugins etwas weiss. Aus Sicherheits-Gründen kann aber ein erfindungsgemässes Plugin am Zielort gewissen, von aussen sicht- oder unsichtbaren Beschränkungen unterliegen. So kann z.B. festgelegt werden, dass ein erfindungsgemässes Plugin eine bestimmte Ausführungszeit nicht überschreiten darf, etc. Wird eine Verletzung einer entsprechenden Beschränkungen registriert, so werden entsprechende Massnahmen ergriffen, indem beispielsweise die Ausführung abgebrochen wird (d.h. das Plugin wird „getötet"). Bei einer Realisierung von erfindungsgemässen Plugins, bspw. mittels der Programmiersprache "Java", wird mittels eines speziellen Class-Loaders am Zielort „auf Befehl" bestimmte Plugins als "Class" geladen. Anschliessend wird davon eine "Instanz" erzeugt, welche dann z.B. über ein Plugin-Interface aufgerufen wird.

Die erfindungsgemässen Plugins werden in der Regel automatisch mittels einer erfindungsgemässen Anordnung erzeugt. Dabei werden in der Regel interaktiv generierte Daten z.B. von Surfsessions verwendet. Bei den generierten Plugins handelt es sich typischerweise um ausführbaren Programmcode. Ein wesentlicher Unterschied zum Stand der Technik besteht u.a. darin, dass die erfindungsgemässen Plugins in der Regel automatisch generiert werden. Ein erfindungsgemässer Recorder, der u.a. zur Erzeugung von Plugins dient, hat vorteilhafter Weise ein Web-Interface in der Art, dass auch ein technisch nicht versierter Benutzer z.B. eine Surfsession aufzeichnen kann, um diese danach in die zentrale Datenbank von Testanordnungen einzubringen, resp. diese als Plugin zu erstellen. Diese Surfsession steht ab dann zur Verfügung um Tests jeglicher Art in periodischen oder willkürlichen Zeitintervallen z.B. durch Sonden auszuführen. Diese bewusste End-User-Funktionalität, die derart konzipiert ist, dass sie ohne technisches Wissen bedienbar ist, bietet zusätzliche Vorteile.

Die Erfindung wird anhand der folgenden Figur näher erläutert. Diese zeigt schematisch ein Netzwerk mit Sonden und einem zentralen Dienst.

Figur 1 zeigt eine vorteilhafte Ausführungsform der Erfindung. Ein erfindungsgemässes Überwachungssystem 1 überwacht über ein Netzwerk (Inter-/Intranet) 2, bei Bedarf von verschiedenen Punkten 3.1, 3.2, 3.3 aus, beliebige Services von einem Host 4 mit Hilfe eines zentralen Dienstes, der bevorzugt mittels einem zentralen System 5 betrieben wird. Test-Konfigurationen, Test-Programme, beispielsweise in Form von erfindungsgemässen Plugins, und auch Test-Resultate werden bevorzugt in einer Datenbank gespeichert, die sich hier im Bereich des zentralen Systems 5 befindet. Auf dem zentralen System 5 läuft ein Programm, welches vorbestimmte oder zufällige Testkonfigurationen periodisch und oder aperiodisch, z.B. zu Überwachungszwecken, oder einmalig über viele Instanzen/Kanäle 10.1, 10.2, 10.3 parallel, z.B. als Lasttest, zur Ausführung bringt. Das zentrale System 5 führt jedoch diese Tests in der Regel nicht selbst aus, sondern übermittelt Test-Programme und Test-

Konfigurationen an eigens dafür vorgesehene Mittel, hier Sonden (Computer) 11.1, 11.2, 11.3. Diese befinden sich vorzugsweise örtlich getrennt in einem Netzwerk 2, z.B. bei Providern, in einem Rechenzentrum, usw. In der Regel geschieht die Übermittlung gleichzeitig an mehrere Sonden (schematisch durch Pfeile 13.1, 13.2, 13.3 dargestellt). Diese führen einen oder mehrere Tests aus und übermitteln ortsabhängige Resultate an ein zentrales System. Hierbei kann es sich um dasselbe oder ein anderes zentrales System handeln. Das zentrale System 5 (oder bei Bedarf auch eine oder mehrere Sonden 11.1, 11.2, 11.3) analysiert und speichert die Resultate und veranlasst gegebenenfalls weitere Reaktionen (z.B. Alarm auslösen). Beim erfindungsgemässen Verfahren wird ein Plugin zur Ausführung gebracht, indem es über ein Netzwerk (2) an mindestens einen Computer (11.1, 11.2, 11.3) übermittelt wird. Das Plugin veranlasst anschliessend den mindestens einen Computer (11.1, 11.2, 11.3), das Plugin zur Ausführung zu bringen.

Durch die erfindungsgemässe Anordnung von einem oder mehreren zentralen Systemen 5 und einer oder mehreren Sonden 11.1, 11.2, 11.3 kann an unterschiedlich (geografischen) Orten im Intranet oder Internet getestet werden, ob z.B. ein zu überwachendes Zielsystem/Server 4 erreichbar und/oder funktionsfähig ist oder ob es gewisse Eigenschaften hat oder ob eine lokale Eigenschaft bei einer Sonde vorhanden und ggf. funktionsfähig ist. Beispielsweise wird ein Web-Server von mehreren Sonden aus überprüft. Dabei wird insbesondere getestet, ob der Webserver von den einzelnen Sonden her, also von unterschiedlichen geografischen Punkten aus, erreichbar ist. Ist der Webserver erreichbar, so wird z.B. auch der „Inhalt" des Web-Servers getestet werden (Verhalten auf HTTP-Requests). Ebenfalls wird bei Bedarf ein Lasttest durchgeführt werden. Wesentlich ist, dass der Server nicht nur von einem Punkt aus, sondern von vielen überwacht wird.

Die beschriebene erfindungsgemässe Systemarchitektur, bei der von mehreren, örtlich getrennten Punkten aus operiert wird, ergänzt mit erfindungsgemässen Plugins, die auf Sonden ausgeführt werden, ergibt ein universelles Testsystem, dass fast jeden erdenklichen Test in einem Intranet bzw. im Internet ausführen kann, ohne dass für unterschiedliche Tests die ganze System-Architektur wieder neu programmiert oder ergänzt werden muss. Es genügt in der Regel, dass ein neues erfindungsgemässes Plugin typischerweise automatisch mittels einem erfindungsgemässen Recorder erzeugt wird und in einer Datenbank eines des zentralen Systems gespeichert wird.

PATENTANSPRÜCHE

1. Verfahren um ein Plugin zur Ausführung zu bringen, **dadurch gekennzeichnet**, dass das Plugin über ein Netzwerk (2) an mindestens einen Computer (11.1, 11.2, 11.3) übermittelt wird und dass dieses Plugin anschliessend den mindestens einen Computer (11.1, 11.2, 11.3) veranlasst, das Plugin zur Ausführung zu bringen.

2. Verfahren gemäss Patentanspruch 1, **dadurch gekennzeichnet**, dass das Plugin aus einer Datenbank von vielen Plugins entnommen wird.

3. Verfahren gemäss einem der vorangehenden Patentansprüche, **dadurch gekennzeichnet**, dass das Plugin automatisch mittels einer interaktiven Surfsession generiert wird.

4. Verfahren gemäss einem der vorangehenden Patentansprüche, **dadurch gekennzeichnet**, dass das Plugin an mehrere, örtlich getrennte Computer (11.1, 11.2, 11.3) übermittelt wird und dass das Plugin diese Computer (11.1, 11.2, 11.3) veranlasst das Plugin gleichzeitig oder ungleichzeitig zur Ausführung zu bringen.

5. Verfahren gemäss einem der vorangehenden Patentansprüche, **dadurch gekennzeichnet**, dass das auf dem mindestens einen Computer (11.1, 11.2, 11.3) zur Ausführung gebrachte Plugin den mindestens einen Computer (11.1, 11.2,

11.3) veranlasst Daten an einen weiteren über ein Netzwerk (2) verbundenen Computer (4) zu Überwachungszwecken zu übermitteln.

6. Verfahren gemäss Patentanspruch 5, **dadurch gekennzeichnet**, dass der weitere über ein Netzwerk (2) verbundene Computer (4) veranlasst wird Daten an einen Computer (11.1, 11.2, 11.3) zu übermitteln.

7. Computerprogramm beinhaltend Computerprogrammcode, **dadurch gekennzeichnet**, dass es geeignet ist mindestens einen Computer (11.1, 11.2, 11.3) dazu zu veranlassen die Schritte des Verfahrens gemäss einem der Patentansprüche 1 bis 6 auszuführen.

8. Computerlesbares Medium beinhaltend Computerprogrammcode, **dadurch gekennzeichnet**, dass es geeignet ist mindestens einen Computer (11.1, 11.2, 11.3) dazu zu veranlassen die Schritte des Verfahrens gemäss einem der Patentansprüche 1 bis 6 auszuführen.

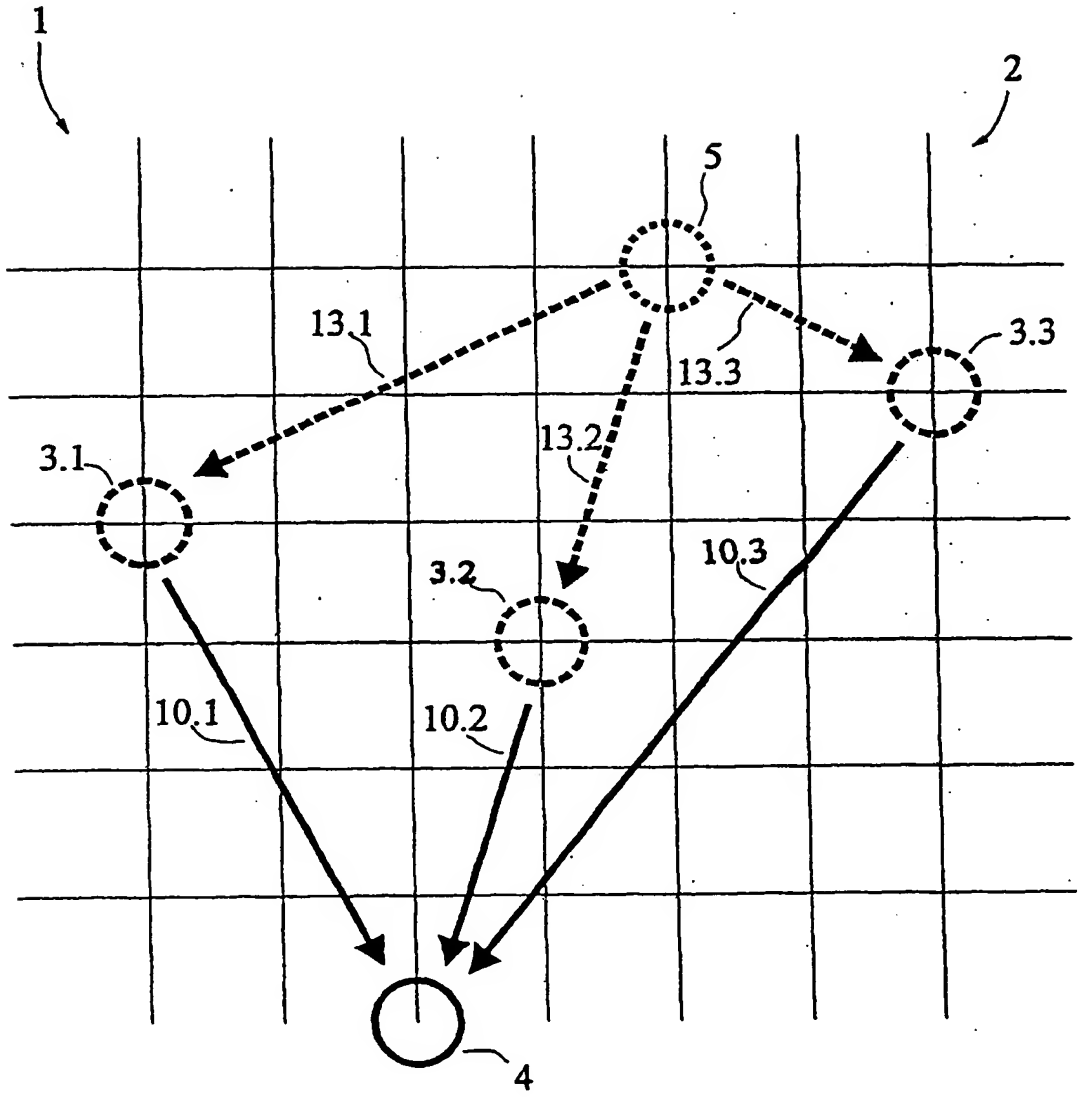


Fig. 1

PF030016
US

(12) International Publication Published under the Patent Cooperation Treaty

(19)World Intellectual Property Organization
International Office

(43) International Publication Date
27 December 2001 (27.12.2001)

PCT

(10) International Publication Number:
WO 01/98899 A2

(51) International Patent Classification[7]:
G06F 9/445

(21) International File Number: PCT/CH01/00373

(22) International Application Date:
15 June 2001 (15.06.2001)

(25) Language in which application was filed:
German

(26) Publication Language: German

(30) Priority:
1222/00 20 June 2000 (20.06.2000) CH

(71) Applicant (*for all destination countries except US*) : **SYSFORMANCE AG** [CH/CH]; Badenerstrasse 281, CH-8003 Zürich (CH)

(72) Inventor; and

(75) Inventor/Applicant (*only for US*): **FISCHER, David** [CH/CH]; Mühlemattstrasse 61; CH-3007 Bern (CH)

(74) Representative: **FREI ANWALTSBÜRO;** Postfach 768, CH-8029 Zürich (CH)

(81) Destination Countries (national): AE, AG, AL, AM, AT, AU, AX. BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU,ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW

(54) Title: Server Monitoring

(57) Abstract
The invention relates to a method for implementing a plugin on one or several computers, in particular for monitoring purposes. The plugin is transmitted via a network (2) to at least one computer (11,1, 11.2, 11.3). The plugin subsequently initiates the at least one computer to execute this plugin.

SERVER MONITORING

The present invention relates to the field of network, or Internet, technology. The aim of the invention is attained by the invention defined in the patent claims.

The Internet has today become widely established as a worldwide communication means. The quality of the offered serves plays therefore a significant role. Companies offering their services on the Internet, have a great interest in the faultless functioning of their servers and in the early detection of unauthorized access and in taking timely measures. Monitoring of these services is not known at this time. For this reason many services offered on the Internet are not at all or only insufficiently utilized. The services frequently are of unsatisfactory quality (response time too long, etc.), which deters potential users from utilizing them. As a rule, unauthorized access and changes are only poorly detected and with delay. This leads to the fact, that harmful software, such as viruses, etc. are undetected and can spread over a relatively long period of time. Destruction worldwide in billions are not rarities.

The aim of the present invention is providing a method for implementing plugins; in particular for monitoring networks, Internet services and servers.

The concept of the invention disclosed here is based on the utilization of a proxy server support of the Internet HTTP protocol for the purpose of automatic recording and later automatic playback of a data traffic from one or several HTTP clients (for example web browsers), which communicate with an HTTP server or HTTP proxy server. Data, preferentially recorded as reference, are stored in a form which permits repeating, monitoring and evaluating the complete data traffic generated by client and server at a later determinable point in time automatically and any desired number of times, in particular from different geographic locations and while maintaining defined criteria. As a rule, the process takes place without any action by the original, generating client. In recording a data traffic conventionally also the response data of the server are recorded entirely or partially. It becomes thereby possible for the first time that at a later application of the recorded client requests, to check whether or not the server supplies analog constant data, or whether it deviates from a defined standard. This is of relevance for the periodic monitoring of unauthorized accesses.

In connection with the monitoring, for example of viruses, if necessary, instead of a most often unsuccessful search for damaged programs, the information is periodically compared with secured and trustworthy reference data (from one or several remote locations). A corresponding comparison supplies meaningful data with minimum expenditures. For example, it is possible that a company offers corresponding services by comparing and

monitoring reference data from individual servers periodically with their instantaneous behavior, for example for the purpose of quality assurance. If needed, the response times of the server are recorded. The monitoring takes advantageously place from different geographic sites, such that the monitoring takes place across several channels. Therewith it is moreover possible to compare and evaluate the performance and discrepancies of the same from definable limit values (in particular across different paths). Alarm messages are accordingly transmitted if required.

The content of the data traffic via a selected protocol (for example HTTP) in the method described here plays a rather subordinate role, i.e. all of the contents can be recorded and played back again even if these relate for example to contents of superior protocols, such as for example JavaScript or SSL. Further application examples of the method described here are, for example, the recording of interactive web browsers surf sessions. It is advantageous to generate one reference session out of one or several sessions. Evaluation and later application of these surf sessions, for example in the form of last-test routines serves for referential monitoring and the checking of unauthorized accesses, as well as the measurement of performance. In particular, the availability of the server is also monitored, in order to monitor hardware defects or crashes. A comparison of the referenced (recorded) server response data with the data traffic generated during an application of the same onto a server, in particular via several channels or paths, is preferably employed as mechanism for detecting modifications of data contents of the server as well as for the location-dependent performance measurement. Illegal accesses and changes are therewith reliably and rapidly detected.

The application purposes of proxy servers provided for example in the currently widely established HTTP protocol are normally *inter alia* the following: intermediate storage of data for the purpose of shortening the response times; logging and evaluating the data traffic between client and server in view of checking the surfing behavior of individual natural persons (observation and checking of person, suppression of undesirable web sites, etc.); prevention of direct connection of individual computers of end users with the Internet for security reasons. The invention disclosed here is remotely based on the functionality of a proxy server. However, in contrast to it, the main function proper of a conventional proxy server is not utilized or only in a secondary manner. In contrast to a conventional proxy server, the invention disclosed here comprises *inter alia* the following differences:

- In order for all data contents between client and server to be recorded compactly in time, in the invention all cache mechanisms (of the normal HTTP logging as well as also of the HTTP proxy log, in particular the references represented directly and carried out by the client), are not included in the consideration (if necessary, utilization can be provided) or are suppressed. For that reason the invention as a rule does not require its own cache.

- In particular, all of the information of the client to the server and of the server to the client are suppressed via cache capabilities in order to attain that all relevant data are transmitted.

The invention comprises means for recording. With interfaces, specifically provided for this purpose, these means are controlled ("start record"). In this state all requests/responses are stored in a defined data structure such that the course of the same can be traced back at a alter point in time with appropriate means (for example an appropriately programmed machine). The data, recorded for example as references, are advantageously deposited in a corresponding library.

If the need arises, plugins according to the invention are generated from these recorded data either automatically or manually (*cf.* in this connection below). The plugins can be implemented via means according to the invention, for example probes (*cf.* in this connection below), such that the same tests can be simultaneously performed in particular from different sites. It is therewith possible to monitor a server with different reference data or several servers with specific reference data. The invention can be applied selectively to one client only to all clients. In the case of clients with a separate recording the client advantageously demands an HTTP authentication. With each request by a client, this authentication can subsequently be utilized to administer the recording data of the individual clients individually.

It is understood that the invention, if required, can also support conversions of HTTP to HTTPS (SSL) or higher protocols. For example, for the recording of HTTPS requests, non-encrypted requests from the client to the server can be made. These non-encrypted requests are in this case only encrypted by the invention and transferred to the server. The response, in turn, is decrypted by the invention and transferred back to the client. It is therein especially advantageous, that the SSL protocol is already decrypted by the invention and not after it reaches the client. It becomes thereby possible to record the data exchange between client and server even if it is encrypted. Higher-level protocols are intentionally broken open for the purpose of recording/monitoring, thereby that, instead of a provided tunneling process, a client-server-client-server process is provided.

Plugins are known in prior art. Plugins are typically universally applicable programs which are specialized to carry out a selected function. To activate a plugin an appropriate plugin interface is required. In the case of Java programs, this takes place, for example, across an appropriate interface. As a rule, the procedure involves loading a plugin due to a request or a demand of a program (for example by a web browser). However, in contrast to the invention disclosed here, with CORBA as well as also with RMI (Java Remote Method Invocation) only data or variables are exchanged, but no program code is transmitted. In the case of the plugins according to the invention, in contrast to prior art, typically the program code is transmitted. In conventional plugins the initiation for loading a plugin always originates from the site at which the plugin is also execute* (from the interior). However, in the case of plugins according to the invention this initiation comes from a different site, thus typically from the outside.

The plugins according to the invention advantageously function as follows: at a first site (original site) at a certain point in time an initiation is started that a plugin is to be executed at a second site (target site) by means of a suitable means. The plugin is thereupon transmitted to a second site (target side) with a request for execution. Consequently, the result comprises that at the second site (target site) a plugin is executed, which for example acknowledges a result back to the first site (original site). The sole requirement at the second site (target site) is that plugins according to the invention can be received and executed (= "starts"), respectively. It is not required that the target site knows anything about the content of the plugin according to the invention. However, for reasons of security a plugin according to the invention can be subject to certain restrictions visible or invisible from the outside. For example, it may be defined that a plugin according to the invention must not exceed a certain execution time, etc. If a violation of a corresponding restriction is registered, appropriate measures are taken in that, for example, the execution is terminated (i.e. the plugin is "killed"). In the case of realizing plugins according to the invention, for example, by means of the programming language "Java", by means of a special class loader at the target site "upon a command" certain plugins are loaded as a "class". Subsequently therefrom an "instance" is generated, which then is called up for example via a plugin interface.

The plugins according to the invention are generated, as a rule, automatically by means of a configuration* according to the invention. Therein, as a rule, interactively generated data, for example of surfing sessions, are employed. The generated plugins are typically executable program code. A significant difference between the invention and prior art comprises *inter alia* that the plugins according to the invention are, as a rule, generated automatically. A recorder according to the invention, which serves, inter alia, for generating plugins, advantageously has a web interface in the manner such that even a technically inexperienced user, can record for example a surfing session in order to be able to introduce these subsequently into the central data base of test configurations* or to establish these as a plugin. This surfing session is subsequently available to carry out tests of any type in periodic or random time intervals, for example through probes. This intentional end user functionality, which is conceptualized such that it can be operated without technical knowledge, offers additional advantages.

The invention will be explained in further detail in conjunction with the following figure. The figure depicts schematically a network with probes and a central service.

Figure 1 shows an advantageous embodiment of the invention. A monitoring system 1 according to the invention monitors via a network (Inter-/Intranet) 2, if required from different points 3.1, 3.2, 3.3, any desired service from a host 4 with the aid of a central service, which preferably is operated by means of a central system 5. Test configurations, test programs, for example in the form of plugins according to the invention, and also test results are preferably stored in a data base, which is here located in the proximity of the central system 5. On the central system 5 runs a program, which executes predetermined or random test configurations periodically and/or aperiodically, for example for the purposes of monitoring, or once via many instances/channels 10.1, 10.2, 10.3 in parallel, for example as

load test. However, as a rule, the central system 5 does not perform these tests itself, but rather transmits test programs and test configurations on means specifically provided for this purpose, here probes (computers) 11.1, 11.2, 11.3. These are preferably spatially separated in a network 2, for example at providers, in a computer center, etc. As a rule, the transmission takes place simultaneously to several probes (indicated schematically by arrows 13.1, 13.2, 13.3). These perform one or more tests and transmit results, which are site-dependent, to a central system. The same or a different central system may be involved. The central system 5 (or optionally also one or several probes 11.1, 11.2, 11.3) analyzes and stores the results and optionally initiates further responses (for example triggers alarm). In the method according to the invention a plugin is implemented, in that it is transmitted via a network (2) to at least one computer (11,1, 11.2, 11.3). The plugin subsequently causes the at least one computer (11.1, 11.2, 11.3) to run the plugin.

Through the configuration according to the invention of one or several central systems 5 and one or several probes 11.1, 11.2, 11.3 it is possible to test at different (geographic) sites in the Intranet or Internet, whether or not, for example a target system/system 4 to be monitored is accessible and/or capable of functioning or whether it has certain properties or whether a local property is present in one probe and optionally whether it is capable of functioning. For example, a web server is tested by several probes. The test comprises whether or not the web server is accessible from the individual probes, thus from different geographic points. If the web server is accessible, the "content" of the web server (behavior upon HTTP requests) is for example also tested. If required, a load test is also carried out. It is essential, that the server is not only monitored from one point but rather from many points.

The described system architecture according to the invention, in which operations take place from several, spatially separated, points, supplemented with plugins according to the invention, which are implemented on probes, yields a universal test system, which is capable of executing any conceivable test in an Intranet or in the Internet, without the entire system architecture for different tests needing to be programmed anew or needing to be supplemented. As a rule it suffices that a new plugin according to the invention is generated typically automatically by means of a recorder according to the invention and is stored in a data base of one of the central systems.

PATENT CLAIMS

1. Method for implementing* a plugin, **characterized in** that the plugin is transmitted via a network (2) to at least one computer (11.1, 11.2, 11.3) and that this plugin subsequently prompts the at least one computer (11.1, 11.2, 11.3) to run* the plugin.

2. Method as claimed in claim 1, **characterized in** that the plugin is taken from a data base of many plugins.

3. Method as claimed in one of the preceding claims, **characterized in** that the plugin is automatically generated by means of an interactive surfing session.

4. Method as claimed in one of the preceding claims, **characterized in** that the plugin is transmitted to several spatially separated computers (11.1, 11.2, 11.3) and that the plugin prompts these computers (11.1, 11.2, 11.3) to run* the plugin simultaneously or non-simultaneously.

5. Method as claimed in one of the preceding claims, **characterized in** that the plugin run on the at least one computer (11.1, 11.2, 11.3) prompts the at least one computer (11.1, 11.2, 11.3) to transfer for the purpose of monitoring data to a further computer (4) connected across a network (2).

6. Method as claimed in claim 5, **characterized in** that the further computer (4) connected across a network (2) is prompted to transfer data to a computer (11.1, 11.2, 11.3).

7. Computer program comprising computer program code, **characterized in** that it is suitable to prompt at least one computer (11.1, 11.2, 11.3) to carry out the steps of the method according to one of claims 1 to 6.

8. Computer-readable medium comprising computer program code, **characterized in** that it is suitable to prompt at least one computer (11.1, 11.2, 11.3) to carry out the steps of the method according to one of claims 1 to 6.